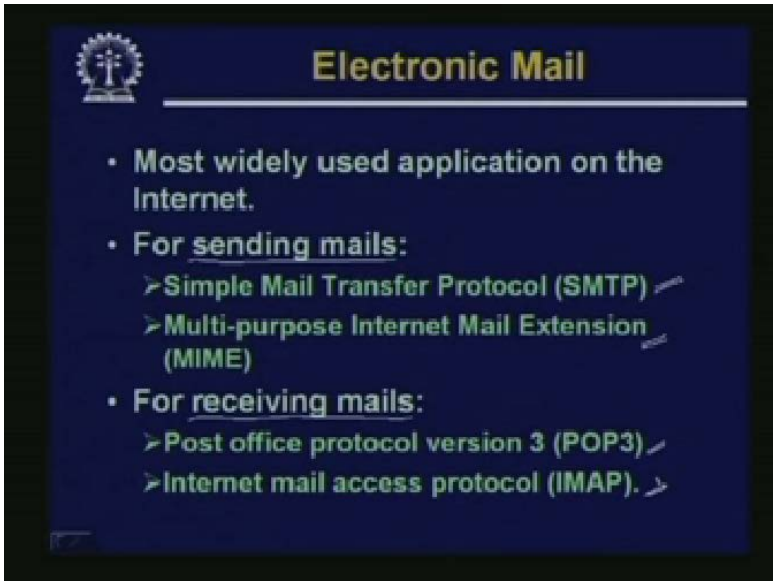
Internet Technology
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture No #10
Electronic Mail

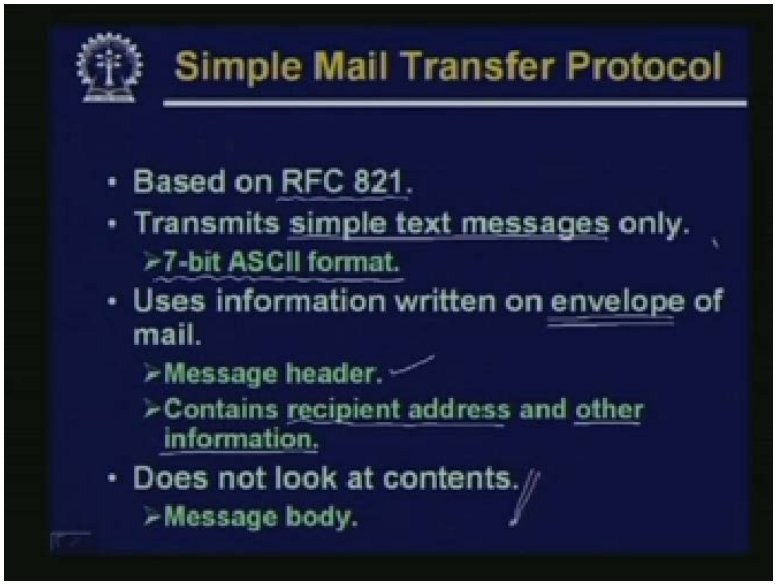
Now, continuing with our discussions on internet applications, today we would be talking about perhaps the most widely used application that people use nowadays over the internet namely electronic mail. So our topic of discussion today is electronic mail.

(Refer Slide Time: 01:11)



Now, you know everybody knows electronic mail is perhaps the most widely used application on the internet. Well anybody who has a connection or connectivity to the internet invariably uses electronic mail or email it is more popularly called in some form or the other. Now talking about email there are a number of underline protocols which are used in practice. For instance for sending mails we use a protocol called Simple Mail Transfer Protocol or SMTP. In conjunction with it we use another protocol called Multi-purpose Internet Mail Extension MIME. On the other hand when you are receiving mails on computers there are two other protocols which are used. In fact one of these two, one is Post Office Protocol version 3 or is it is also known as POP3 in short or Internet mail access protocol or IMAP.

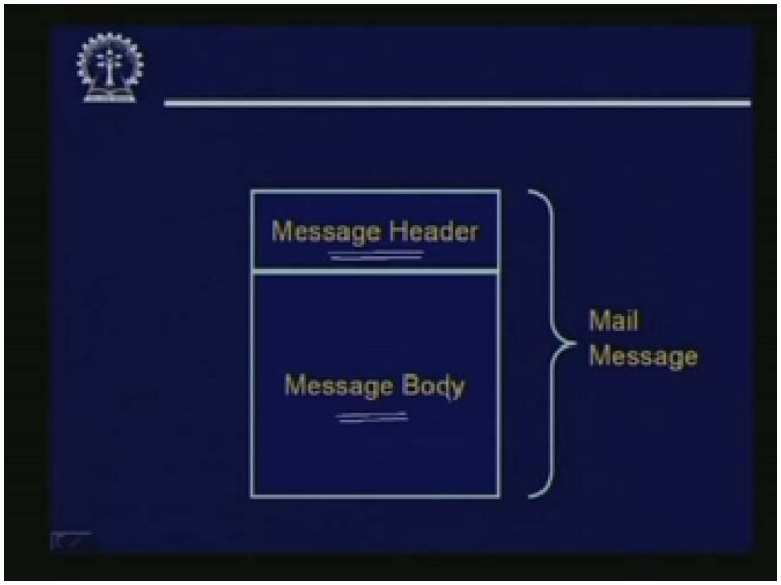
(Refer Slide Time: 02:16)



Now let us start our discussion on Simple Mail Transfer Protocol. In fact this SMTP forms the basis for all electronic mail transaction that takes on the net on the internet. Now this SMTP is the basic protocol used to send mail from one machine to other. So let us try to understand what this SMTP protocol is? What are its capabilities and limitations and how it works? The first thing is that you can get all the details about SMTP in RFC 821, this is the corresponding RFC. Well SMTP is a simple protocol in the sense that, it has the capability of transmitting simple text messages, only it cannot transmit other than simple text. Well when you say simple text it means a simple seven bit ASCII file. You cannot have any international characters which may require 8 bit ASCII encoding. It must strictly be 7 bit ASCII format.

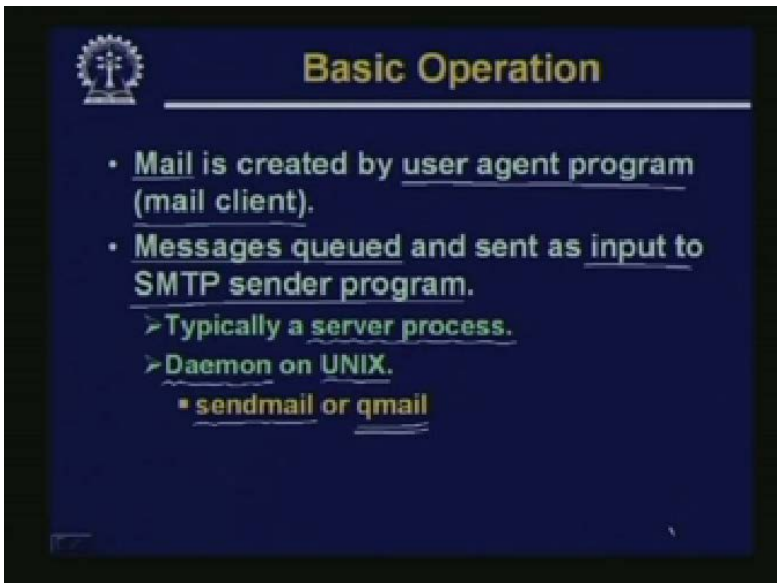
But the mostly the significant bit of the bytes will be always 0. And in the SMTP protocol, the mails are transferred based on information which is written on the envelope of the mail. Just like when we send a letter to a destination, we put the letter inside an envelope and on top of the envelope. We write all the details namely the address which will help the post office to deliver the letter at the correct destination. So in the same inside the envelope you have the Message header. Message header can contain the address of the recipient and some other information this we shall see later what other information are required. This SMTP while it is transferring or transmitting mails it does not look at the mail contents or the message body as long as it is in simple text format it only look s at the message header. That is only important to send the mail to the correct destination.

(Refer Slide Time: 04:34)



So effectively a mail message look s like this there are two parts. One is the Message Header, other is the Message Body.

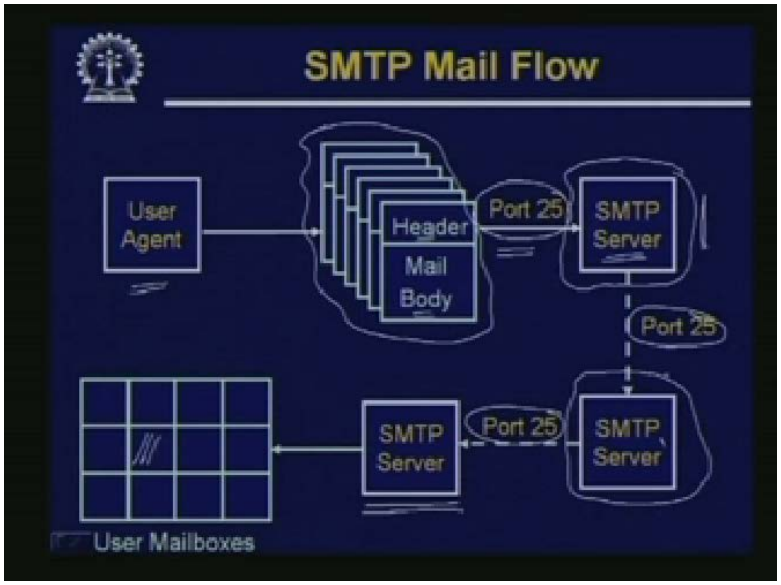
(Refer Slide Time: 04:44)



Basically when you are trying send a mail, the Mail is created by a so called user agent program which is also sometimes called as a mail client. Some examples of usage and programs are Outlook Express Outlook or you can say Pine Eudora this kind of mails you can say client or user agents are available. Based on the platform you are using you can use one of these platforms to compose your mail whatever you trying to send you can type in the body of the mail. Now once I have created the mail and have given a command to send it, the messages get queued up and these messages go the input of an SMTP sender program. We shall be illustrating this by an example. With the help of a diagram now this SMTP sender program is actually responsible for sending the mails. This SMTP sender program is run

usually as a server process which for example in the UNIX environment it is sometimes called a Daemon process. Some examples of some SMTP process are sendmail or qmail. Both of these are widely used qmail is slightly better than sendmail because of some of its attractive features.

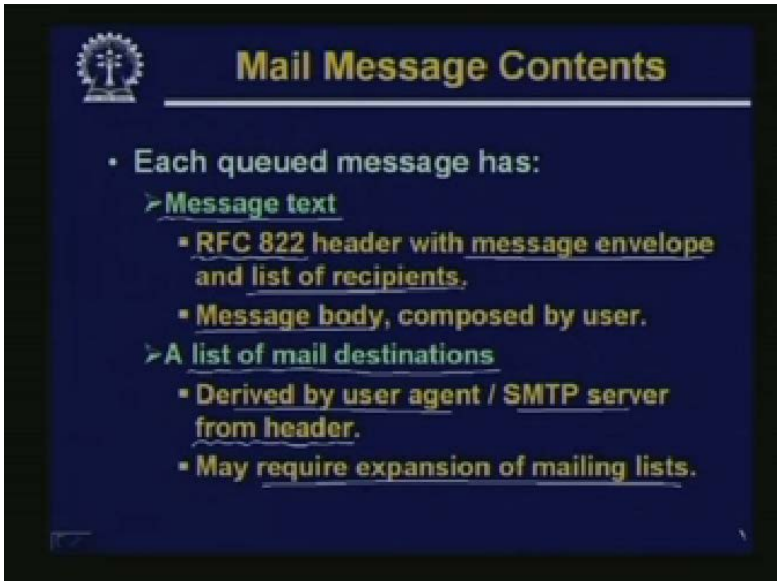
(Refer Slide Time: 06:14)



Now let us try to illustrate the process with help of this diagram. Here we have the User Agent where the mail is getting composed. But however this User Agent may not be running the SMTP server program. Suppose this is the place where the nearest SMTP program is running out here. So this SMTP server will be acting as the SMTP sender or the mail sender. The User Agent after the mail gets composed will open a connection to the SMTP server on Port number 25. Port number 25 is the Port for SMTP server. And as you can see several mail messages might get queued up in the User Agent (07:08 audio not clear) processed by the SMTP sender program, so each of these queued up mails will be containing Mail Header followed by a Mail Body. Now this SMTP server program when it receives the mails, it will try to forward the mail to the destination. Now it is possible, but it does not have a direct path to the destination it may have to send the mail through a number of intermediate mail servers this may be necessary.

So for that purpose mail server program out here this may send the mail to some intermediate SMTP server. One or more of them with a connection on Port 25 established in each case and finally it will be delivered to an SMTP server which has a connection with the destination User Mailbox directly. So this SMTP server will receive the mail and will find that well this particular mail is destined to a mailbox on my machine. So the mail will get deposited in the appropriate mail box. This is how the overall thing works. So starting from the User Agent you go to an SMTP server which acts as the SMTP sender. To start with the SMTP sender will be sending the mail to another SMTP server which may simply forward it to another SMTP server and this process will go on till the mail receives the final SMTP server which acts as the final destination or receiver which accepts the mail and puts it in the User Mailbox. This is how the overall SMTP mail transmission phase works.

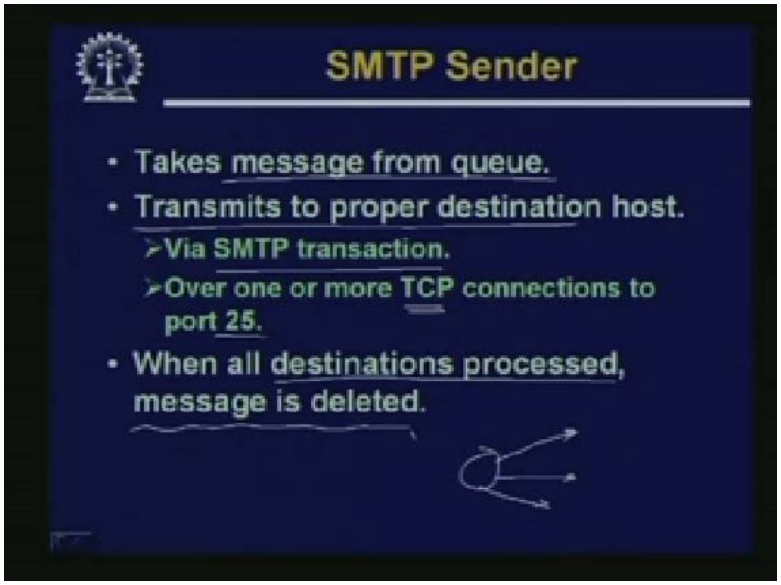
(Refer Slide Time: 08:56)



Now in terms of the mail message contents, each message that gets queued up contains of course the text of the message. The message text in fact contains a header and body. Now in the header there is an RFC 822 which defined the fields of header. So if you look at 822 you can know all the details of the header fields. So this header will contain message envelope containing some information as we shall see in some examples and also a list of recipients. And it will also contain the message body which of course will be composed by the user. And in addition to the message text each queued message will also have a list of mail destinations. This list can be derived by the user agent or the SMTP server from the header. The idea is that although the mail destinations are anyway present in the header of the message.

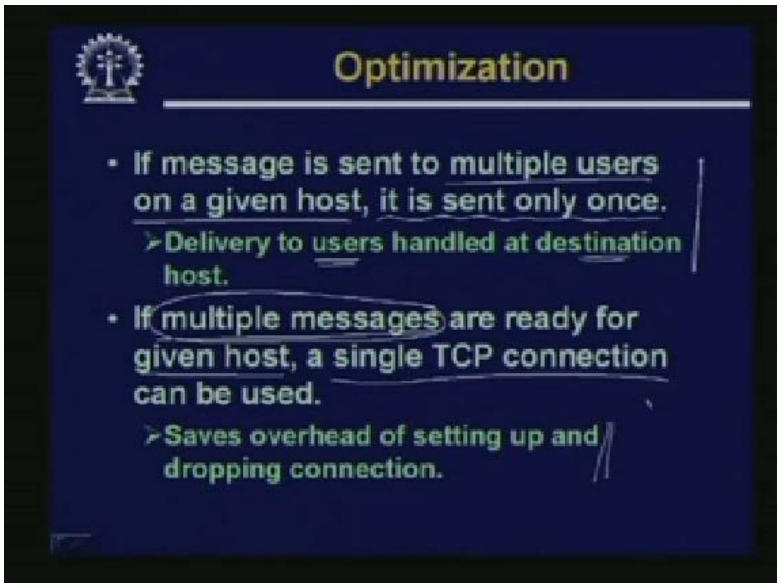
But in order to process it in order to forward it that same information is extracted from the main header and also kept in a separate place which can be accessed very fast by the mail sender program. So the mail as a whole is kept somewhere and only the destination addresses are kept separately along with a pointer to the corresponding mail. So that it can be processes relatively fast. And this list of mail destination may sometimes require expansion of mailing list. Like you know sometimes we have the group mail facility we give an email address which actually is a mail alias which actually refers to a group of email addresses. So when a mail is being sent to that alias, that alias expands to all the email addresses that belong to that group. So this kind of expansion can also take place.

(Refer Slide Time: 10:50)



With respect to the SMTP Sender it takes messages from the queue one at a time. It tries to transmit it to proper destination via an SMTP transaction. It is a TCP connection over port number 25 and when all the destinations corresponding to a mail has been processed please note that a particular mail message may have more than one destination. Unlike a postal an electronic mail can have more than one destination. So when all the destinations have been processed, there are by the copy of the mail has been sent to all the destinations it is destined to the messages deleted from the queue.

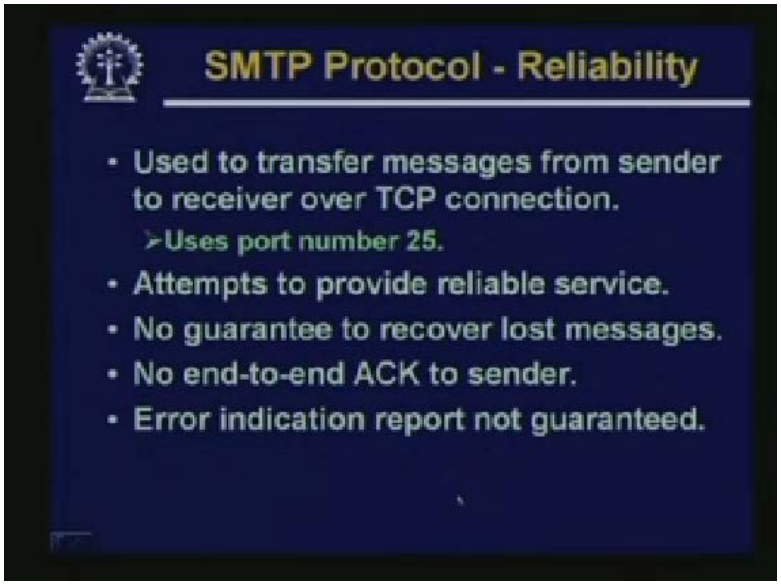
(Refer Slide Time: 11:34)



Well some optimizations are carried out by the mail server like, if a message is being sent to multiple users on the same host, then the mail message will be sent only one to the destination host. The delivery to the individual users will be handled at the destination host. So if the mail is going to two different users on the same machine Hotmail.com. Say only a

single copy of the mail will go to Hotmail.com with information in the header that this mail has to be delivered to two of these addresses. So the final destination say Hotmail.com will do the rest after receiving the body. And secondly if there are multiple messages are ready for the given host may not be the same user, multiple users then instead of sending. Well this is not the same message in the first case it was the same message going to more than one user on the same machine but here multiple messages going to the same machine. Then to save time a single TCP connection can be opened and all the messages can be sent. This actually overheads this actually reduces the overhead of a creating and terminating. The TCP connection over port number 25 you have to do it every time. We are sending mail, so it reduces the number of time you are establishing the connection.

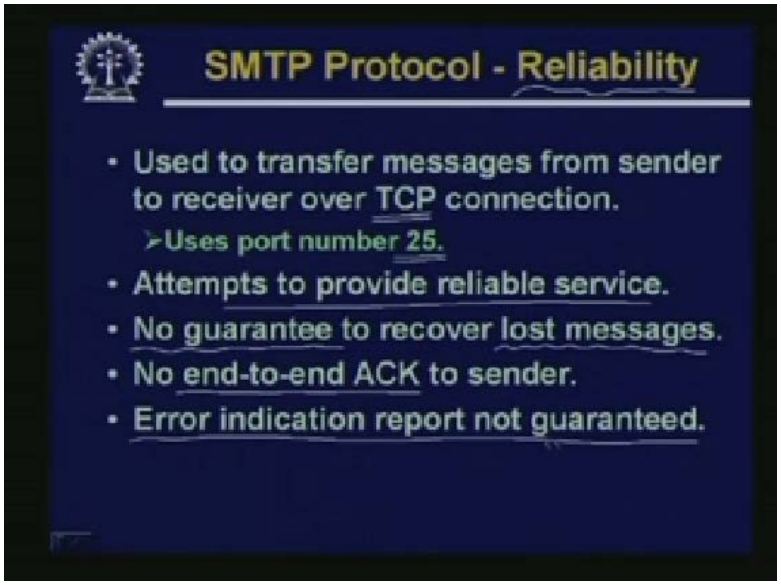
(Refer Slide Time: 13:04)



Now while sending a mail there are a number of errors which are often encountered. Like due to some problem a host may be unreachable. It can be a problem with the host; it can be a problem with the link, through which the host is connected. Due to some reason the host of maybe out of operation. TCP connection may fail due to some error in the network. The destination address may be faulty like the name of the user or the user id may be wrong. The target user address may have changed. Sometimes some of the mail servers allow redirection. If the user address has changed automatically the mail will be redirected to an alternate email address. Now if this redirection is not possible usually the user is informed in the form of an error message. And if this redirection feature is there then the sender can re-queue the mail well after the message comes back from.

The final sender final mail server now which of course it can keep in the queue for certain period. After that it will give up because if the mail has failed often a mail bounces with an error message. That well I have kept it in my queue. I am unable to have a connection or some problem. Now the mail will remain on an outgoing queue for certain period of time that particular period of time is considerable. It can be 1 hour, it can be 1 day, it can be 7 days, and it can be even more. Depending on the time the SMTP server will be keeping on trying to send the mail during that period. And if it fails then that period will simply discard the mail after that.

(Refer Slide Time: 14:50)



Talking about the reliability of SMTP protocol. Well we have mentioned that it uses TCP connection over port number 25. Well we know that TCP is a connection oriented protocol it tries to recover from errors. Supposedly TCP is reliable but in an application like electronic mail which runs on top of TCP there can be so many different kinds of errors. Like I just mentioned some host may be unreachable, some user email address may have been redirected, some other address and so on. So under these conditions, so all though TCP attempts to provide reliable service, there is no guarantee that the mail will be delivered directly. Some messages may be lost. There is no guarantee to recover lost messages see although tries to ensure that end to end transactions successful.

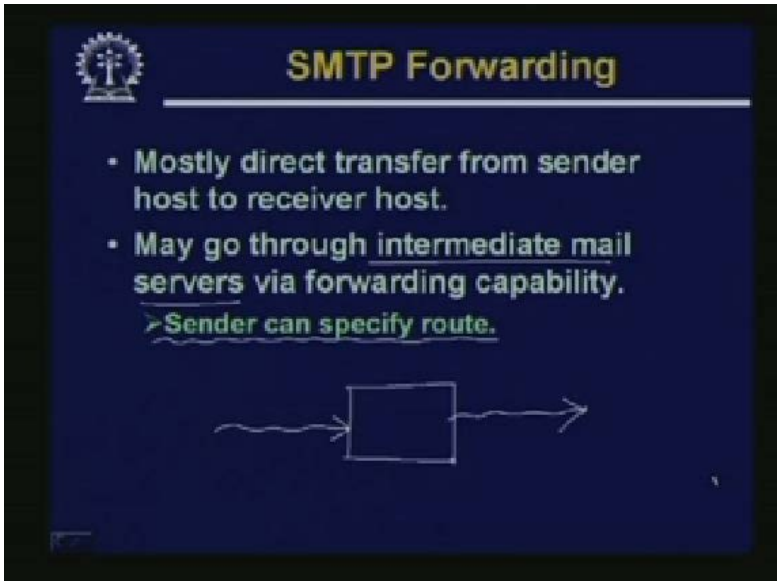
For example from your machine to the nearest SMTP server that transaction is successful. But somewhere beyond that something might have happened which has called the mail message to get lost. So there is no effort made by the SMTP protocol to recover from these lost messages. Similarly, there is no end-to-end acknowledgement from the final destination back to the sender that is used in SMTP. Similarly error indication report is not guaranteed sometimes you get back a message that your mail has bounced. But even if you do not get back a message there is no guarantee that the mail has actually reached the destination. We trust SMTP well if you do not get an error message we assume that the mail has reached the destination. But in practice it may not be so.

(Refer Slide Time: 16:50)

Well, now at the side of the receiver well when you are talking of SMTP receiver we were talking of the SMTP server at the final destination which is finally connected to the user mailboxes. These are the user mail boxes. Now these SMTP receivers will be accepting the arriving messages and they will be placing the messages in the appropriate user mail boxes. If the mails are destined to a mail box connected to the particular server. But if it finds that the destination address does not belong to a particular user in the mailbox of its own, machine own server. Then it copies the mail message to an outgoing queue to some other SMTP server to a process called mail forwarding. Suppose I am an SMTP server I receive a message, first I see if the destination corresponds to a user on my machine or not. If so I put in the appropriate mailbox.

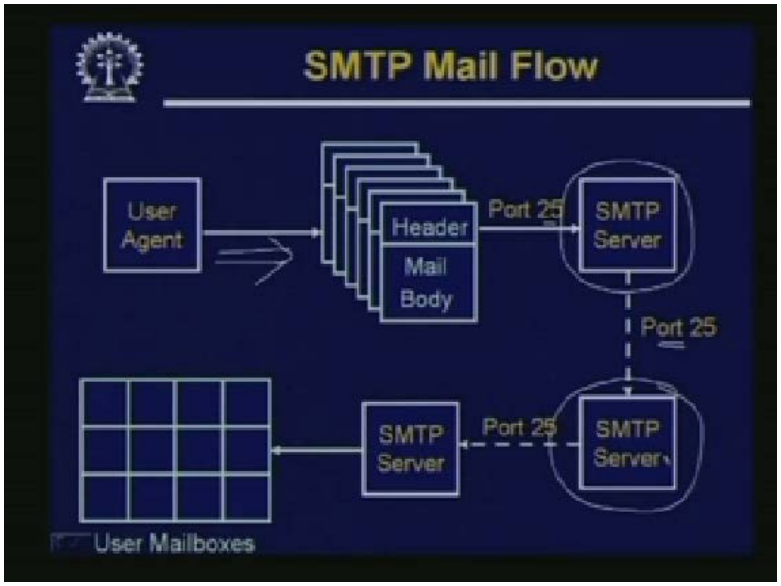
If not, I try to find out which SMTP server I have forward it next I simply forward it to the next SMTP server. This is how it works. So in doing so the receiver must verify the local mes mail destination of course. Because it will have to verify before actually putting them in the mailbox and it deals with some kind of headers like. It can check whether the packet that has arrived, has any error or not. There is a checksum computation or it can also check whether the user mailbox where you are trying to put the mail whether it has some free space or not. Many mail servers allow something called a disk quota where the maximum amount of space you can use for storage of your mails is limited. And if that quota gets exceeded or filled up, no more new messages can be put in there so that is also an error condition.

(Refer Slide Time: 19:02)



And SMTP forwarding means you have an SMTP server out here. It gets a mail as input it finds out. It has to forward the mail to some other SMTP server. This I a process called forwarding. Now this forwarding may need to go through a number of intermediate mail servers in general. Now it is possible in electronic mail in email that the sender can specify the exact route. Well if it does not then like routing each SMTP server will try to find out the best route. Well the sender can also specify a route through the mail messages will go the sequence of SMTP servers the mail message will traverse before reaching the final destination mail server.

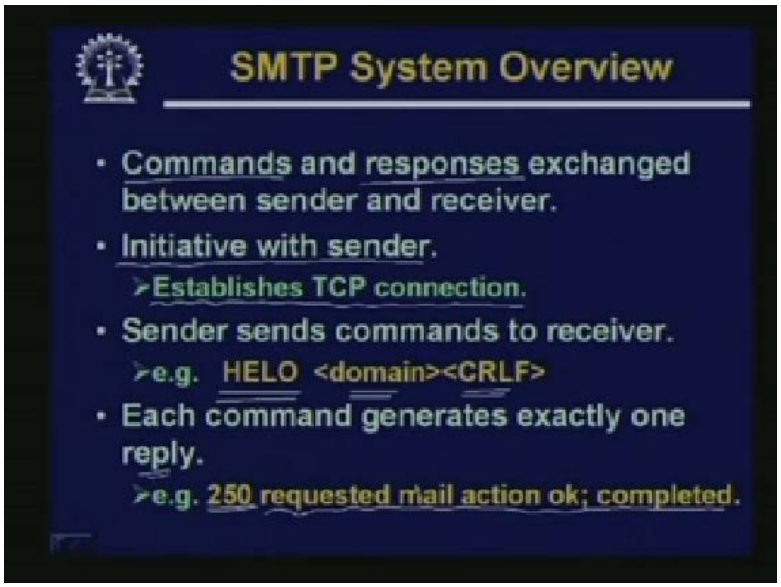
(Refer Slide Time: 19:56)



So again looking back at the same diagram I want to just make a one comment here. If you look at the first SMTP server in this chain. The SMTP server which receives mails directly from the User Agent. This works as a SMTP receiver and sender both it receives mails from

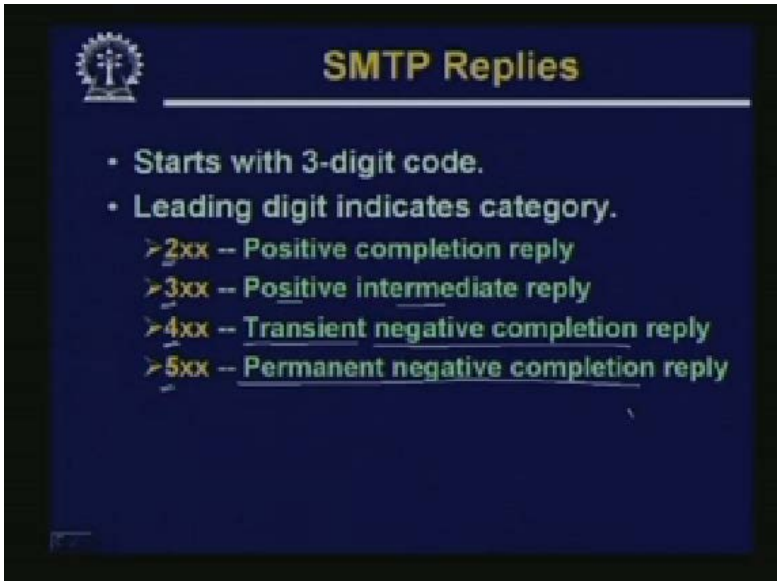
the User Agent over Port number 25. It forwards some of the mails again over Port number 25 to some other SMTP server. So what I mean to say is that the intermediate SMTP server they have dual functionality. They act as server when they are receiving mails from somebody else. They also act as SMTP client when they are forwarding mails to some other SMTP server. So you see these. These SMTP nodes intermediate nodes they have a dual role of both client and server. They are client when they are trying to send the message to another server. They are server when they receiving a message from another node which is acting as the client. So this is what we remember.

(Refer Slide Time: 21:16)



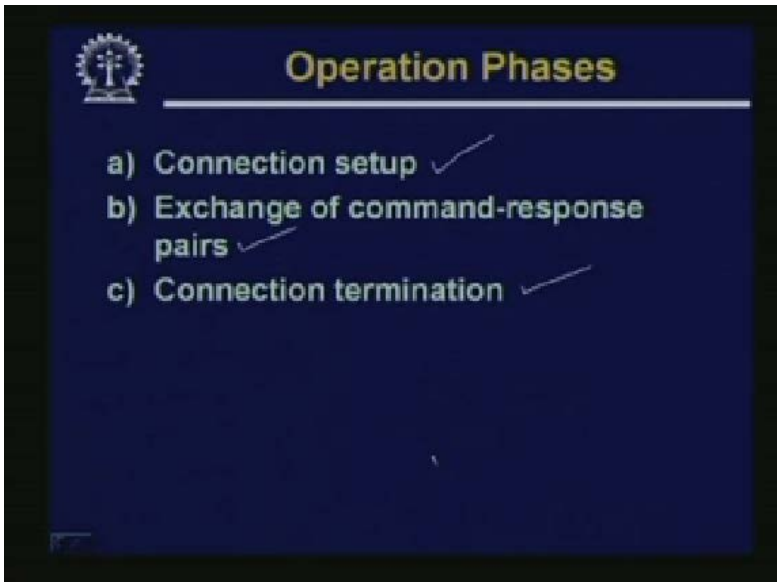
Now talking of the basic commands that SMTP uses well there are commands, there are responses. One node sends a command to the server; a client sends a command to the server. The server sends back a response to the client. Now let us see what this command and response messages look like for they actually mean and represent. The first thing is that this command and response the initiative lies with the sender. Because if the sender is not responsive this dialogue cannot go on the process starts by the client establishing a TCP connection with the SMTP server of course over port number 25. There are a number of commands we shall see through a complete example. Sender can send some command to the receiver. An example command is HELO. In one such command HELO followed by a domain name followed by carriage return and line field this is a command. And the server after receiving the command the reacts by responding with a reply. A typical reply look s like this it starts with a 3 digit response code followed by the actual string indicating the meaning of this code. For example 250 requested mail action completed.

(Refer Slide Time: 22:49)



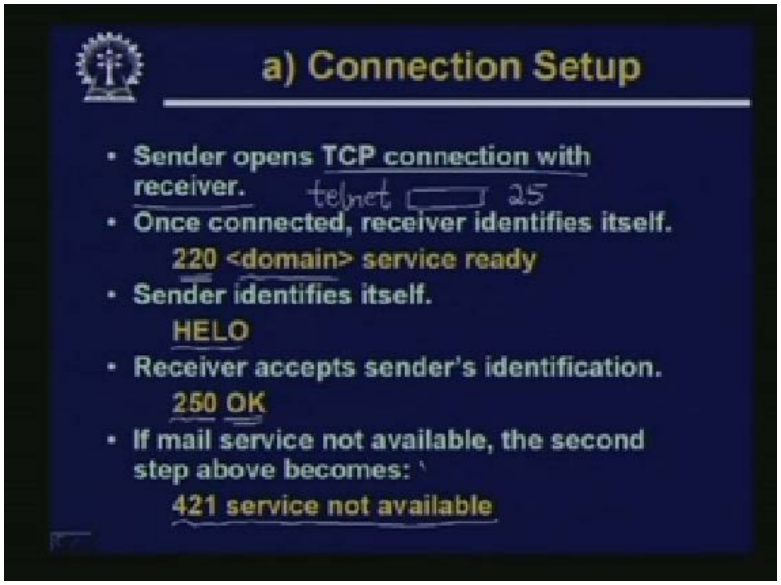
Talking of the 3 digit codes. These 3 digits code can start either with 2, 3, 4 or 5. Well any numbers starting with 2 means Positive completion. That means it is a positive response it is successful and the step that it is referring to has been completed successfully. Well anything staring with 3, this also means positive. But this is not the end of the story. We are only half way through the process this is sometimes like a positive intermediate reply. You may have to go through one or more such positive intermediate reply. So as to reach or arrive at a positive completion reply. Similarly anything staring with 4 is a Transient negative completion reply negative completion means some failure. But Transient means well we are not very sure about what it is about yet maybe this error will be recovered. But 5 starting with 5 means this error is of a permanent nature and the mail transaction has to be stopped or aborted these are meaning of the codes.

(Refer Slide Time: 24:02)



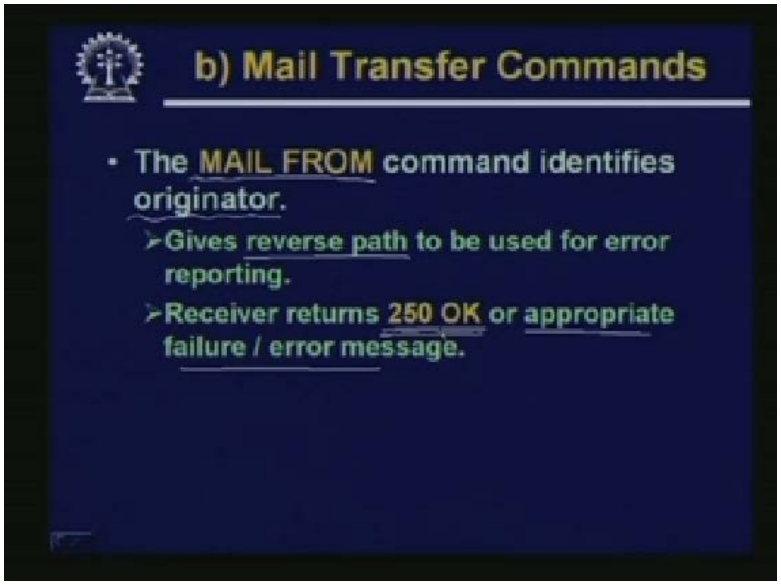
Now in terms of the dialogue that goes on between the mail client and the mail server broadly. There are three steps. First is the connection setup. Second is a step where a number of command and responses are exchanged between the client and the server. And finally the connection gets terminated. Now let us see how these three steps look like.

(Refer Slide Time: 24:26)



Well in the connection set up step the sender opens a TCP connection with the receiver. This is very easy this can be done by the sender giving an explicit telnet command. Telnet followed by the server name whatever server you are using, followed by the port number of SMTP. Telnet server name followed by 25, this will allow the client to open an SMTP connection with the SMTP server. Now once connected the receiver will identify itself by message like 220, its domain name server ready. Sender will then identify itself by HELO message. You will see a complete transaction very shortly receiver will accept the sender identification by sending back a 250 positive reply. Now if the mail service is not available at that particular point. Then instead of 250, a message like this will come 421 indicating this is a negative completion reply, 421 services not available.

(Refer Slide Time: 25:46)

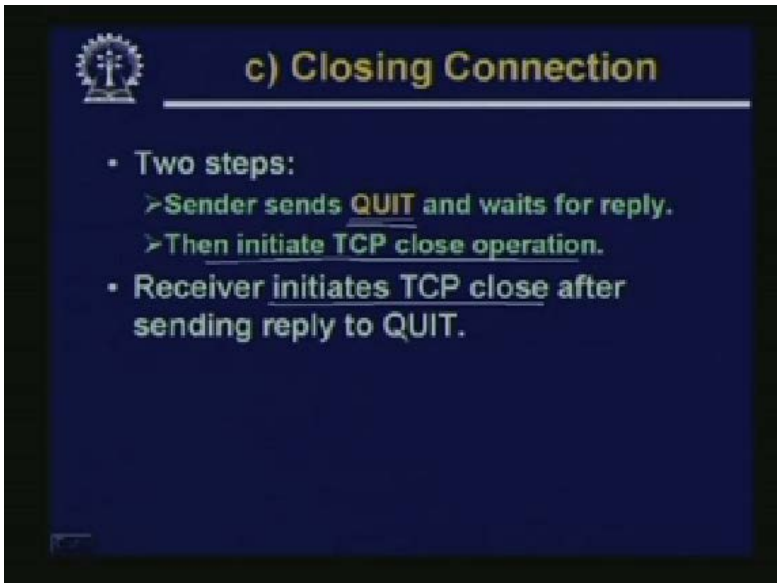


Now in the second step when we are talking about the actual mail transfer commands, there are some commands which identify the email address of the source and the destination because email of email address of the destination is required for obvious reason the mail has to be delivered sometimes email address of the source is also required particularly when error messages have to be sent back in case of some errors the receiver might like to send back an email message notifying the error back to the sender. So for that reason the email address of the sender is also required to be present as parts of the email message the header. So in the mail transfer command these kinds of messages are used. There is a command called MAIL FROM command which identifies the sender of the email or the originator. This as I had said this is required to provide the reverse path for error reporting. Suppose you have to send back an error message you can use this email address for doing that and if the email address, if it has the proper syntax this the receiver will return with a 250 message or if there is something wrong then there will be an appropriate failure message.

(Refer Slide Time: 27:14)

Similarly there will be a receipt TO command where you can identify the email address of the recipients. There can be more than one recipient which is identified by more than receipt TO commands. There will be reply for each recipient. Finally there will there is a command called DATA which indicates that now the actual body of the email starts. So after the DATA command the client can start sending the messages which will form the body of the email to the mail. You can say to the SMTP server for processing and this message can go on continuously. The end of the message will be indicated by a line containing just a full stop or a period. A full stop appearing in the first column of a line will indicate the end of the message. Before that you can go on typing whatever you want.

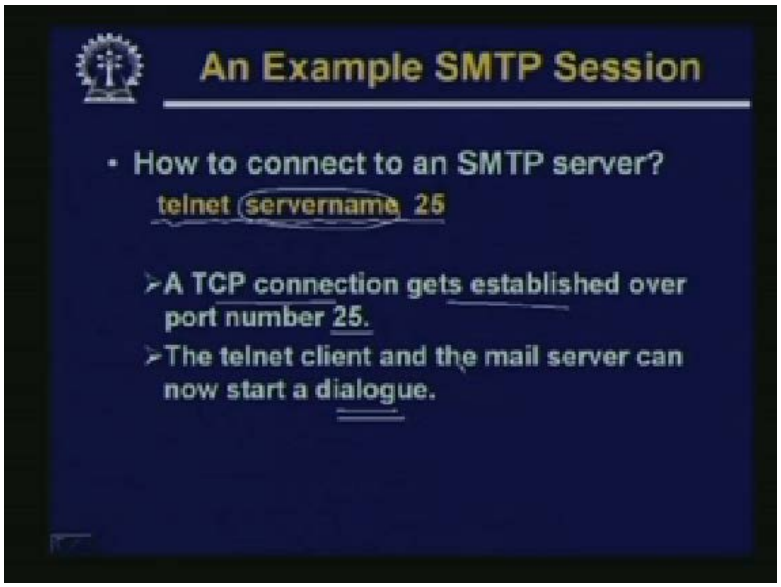
(Refer Slide Time: 28:17)



And finally when you are trying to close the connection, there are two steps. Basically the sender or the client will be sending a quick command to the server and will wait for the reply.

After the reply comes back from the server it initiates TCP close operation. The receiver similarly after sending the response to quit. It also initiate TCP close. This is how the connection is closed.

(Refer Slide Time: 28:48)



Well, now we try to demonstrate an example SMTP session. Well normally the way we send and receives mails is that we use some mail software look Outlook, many of us also use a web based like Hotmail, Rediff Mail, and Yahoo Mail. They have a suitable web based user interface through which we type mails we shall see later. How these are implemented in practice. See at some point in the implementation, these machines where you are sending your mails the body of the mail addresses subject whatever to they must respond to the SMTP request and response. There must be an SMTP server situated somewhere which must be contacted to send. The messages we have typed as an email message.

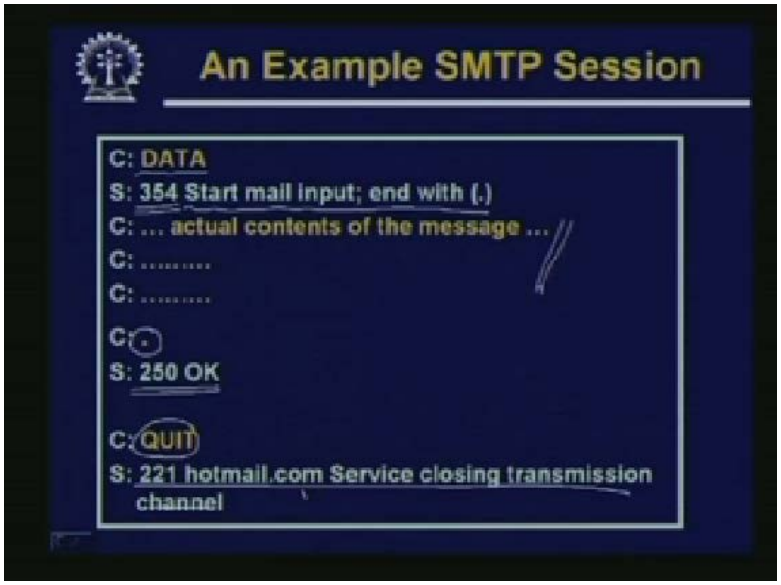
So this simple example that we will show, will demonstrate how a low level SMTP session is carried out. Well although we may be using a high level tool or a high level package to send the mail. But actually in internally these are the kind of commands that get generated automatically and your mail gets sent. So let us see how the process starts. Here I am assuming that the client is a simple window a command window from where I am establishing a connection with my server. Well this can be any mail server I am giving a hypothetical name, server name. If you give telnet over port number 25, this will open a TCP connection over port number 25. So a TCP connection gets established over port number 25. Now after this connection is established the dialogue can start.

(Refer Slide Time: 30:45)

Let us see the slide shows you how the dialogue goes on. Here S indicates the mail server client indicates the mail client. So after the connection is established the server responds by a welcome message 22. Suppose this server is Hotmail.com server I am connecting it to Hotmail.com, it will send 220 hotmail.com Simple Mail Transfer Service Ready, something like this. Now after that the client will give a HELO message and also it will identify the domain name. The client belongs to suppose the client belongs to the domain Yahoo.com. It will respond sending HELO Yahoo.com. The server will again be sending back a positive response that well HELO Yahoo dot com I acknowledge you. This means that the server has identified or has registered the domain name from where the client is sending.

And again the client uses the MAIL FROM command to identify the sender of the mail. Here you can specify any add email address you want. Well when I say any email address it is actually any email address you can also write in place of the email address that we will specify. You can even write a say for example Bill Gates at Microsoft dot com you can write anything out there. So if the syntax is correct the server will respond by 250. Similarly after that there can one or more receipt TO command. In fact in this example there are two where you can specify the email addresses of the recipient. Again the server will be responding by 250 message. So now the sender and receiver of the mail has been identified.

(Refer Slide Time: 32:42)

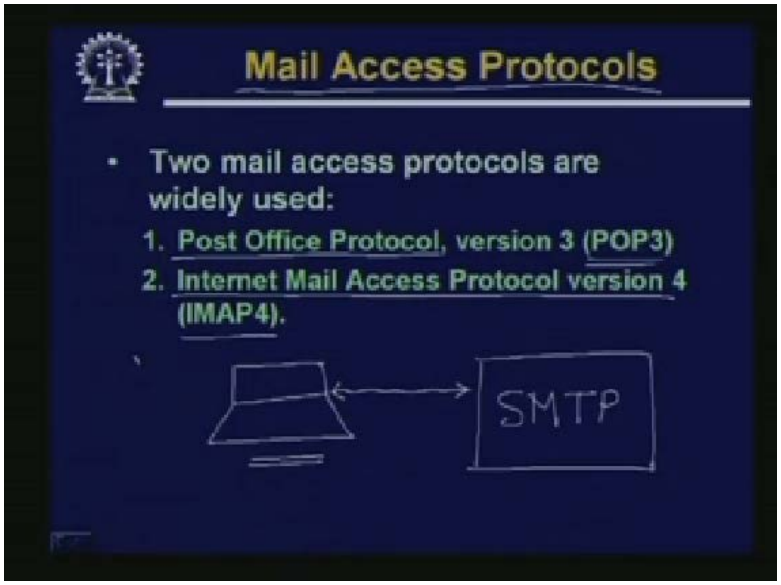


So now let us (doubt) so now the client sends command DATA. Server sends back an intermediate positive response 350 saying that start mail input end with dot. Depending on the server this mail message may differ slightly but actually the meaning is the same. This is intermediate because the mail transfer is still in progress; the body of the mail is still being typed. So it is just an intermediate message asking the user to type in the message and ending it with a full stop or a dot in the first column. So after this the actual body of the mail can be typed and finally a single dot in the first column will indicate the end of the mail message. So at this point the server will respond by a positive response 250.

Then the client can send QUIT and the server will finally send this message and close down the connection. Now the point to note here is that this is a simple mail dialogue I have shown in fact anybody can use this command to have a connection with the mail server which is available to you. For example in your organization if you have a mail server, you can directly do a telnet to your mail server over port number 25 and you can send a mail message by fabricating any arbitrary source address. Well if somebody is not that much familiar with how this email headers and other stuff look like, they might get confused by receiving a mail. For example if I get a mail message from a say for example Bill Gates.

But without knowing that it is really from Bill Gates, but some of friend has played a prank on me. It is quite possible but if I look at the email header carefully I will know the mail although it is saying it is coming from somebody at Microsoft dot com. But in fact the mail originated from some other place. In fact today when you have so many email spam filters which are supposed to block and identify spam mails. They will in fact detect these kind of you can say mischievous mails. Let me call it, but otherwise the SMTP does not check for these kind of what should I say misuse, misuse of mail of transfer facility.

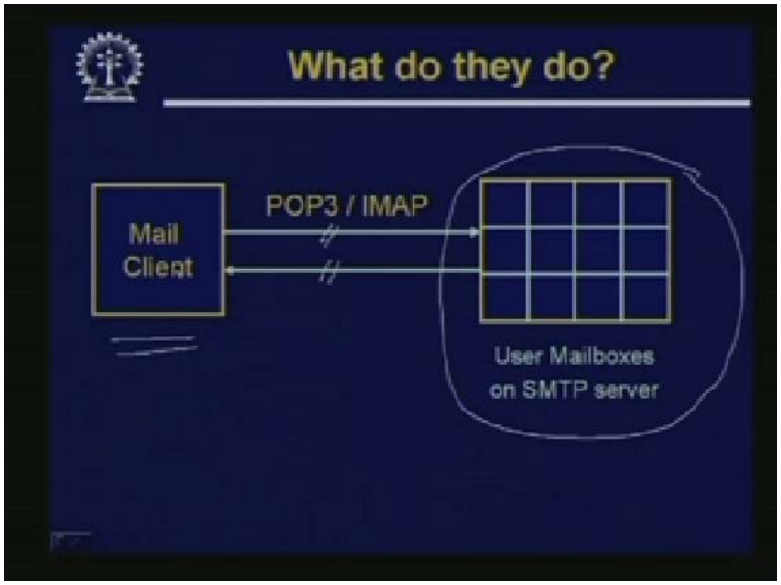
(Refer Slide Time: 35:24)



So far we have talked about how mails can be sent from one machine to other using SMPT. Now we are talking about mail access protocol. See here the idea is like this you have an SMTP protocol running on a machine out here. This is your SMTP server, but I am sitting on a computer out here. And here I may be having a software like Outlook Express. I can establish a connection with the SMTP server. And through this connection I can access, I can read and I can send mails sitting on my computer. Only I need not have to login into this SMTP go to the SMTP server and check my mails. There are in fact two protocols which allow you to access your mails from a remote mailbox.

The idea is that you have a mail server located somewhere in the network. Your mailbox is present on that mail server, you have a mail client, that mail client should have a capability of pulling your mails from the mail server into your machine. So that sitting on your machine you can view the contents of your mails. In fact there are 2 protocols which are used for this purpose. One is called the Post Office Protocol version 3 or POP3 in short; the other one is Internet Mail Access Protocol version 4 or IMAP4. These are the two alternate protocols available. Let us look briefly at the capabilities of these.

(Refer Slide Time: 37:04)



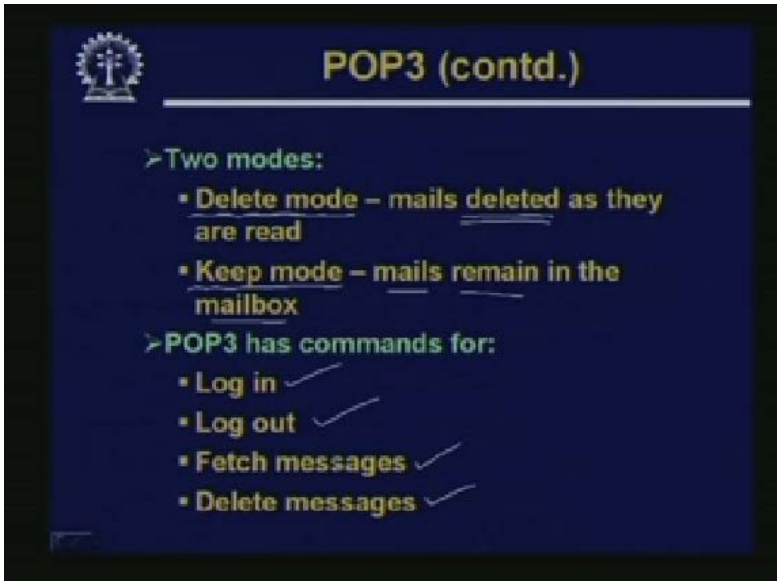
The way they work as I had mentioned is like this. This refers to an SMTP server mailbox. This is your Mail Client. So whenever the Mail Client wants to read mails, it sends a request to the SMTP server to in fact not the SMTP server, to the machine which is running the SMTP server and will be sending back the response in the form of the mails and other information you want to have a look. Now in fact the SMTP server is one program which is running. Suppose when you are trying to access a mail through POP3, you must in addition have another server the POP3 server running on the same machine as the SMTP server.

(Refer Slide Time: 37:54)

So first let us talk about POP3. So for POP3 to work the client, POP3 software must be installed on the recipient machine. Well in fact Outlook Express and the other mail client they have an in-built feature of specifying which mail client you want to have. And the server POP3 software is typically installed on your mail server, the same machine which runs the

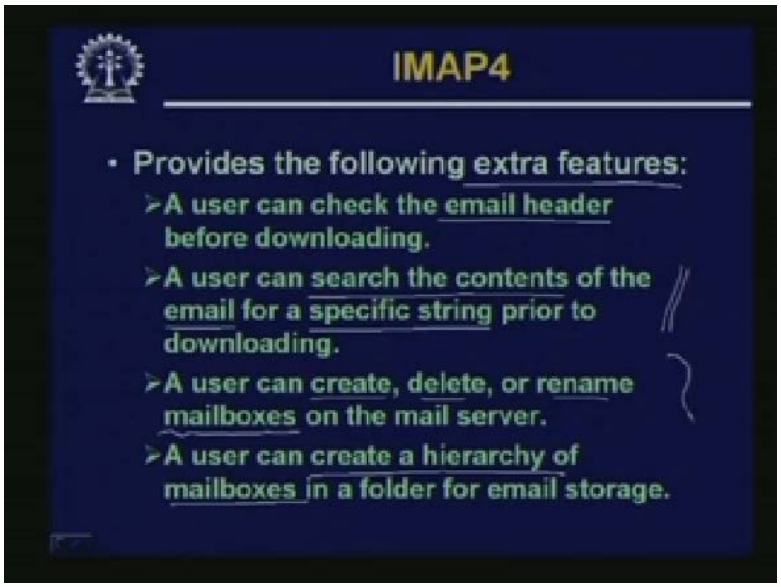
SMTP server. So the client which is user agent here opens a connection with the server on TCP port number 110. This 110 is the port number of POP3, then the client authenticates (38:34 audio not clear) server by sending a valid user name and password. And once the verification of the user name password has been done the client can directly have an access to the mails sequentially one by one. So the mails can be read by the client by transferring them from the mail server one at a time one by one sequentially.

(Refer Slide Time: 39:02)



Now when the mails have been read sequentially one after the other, there are two different modes in which you can do so. One is the Delete mode. Delete mode says that as you are reading the mails they are getting deleted from a mail server so that you are not unnecessarily occupying the disk space of a mail server. You are freeing the disk space as soon as you have brought your mails to the machines for reading. But the other option is that if you do not want to do so, there is Keep mode where you can leave the mails intact in your mailbox. So POP3 has basically commands to Log in, Log out, Fetch messages without deleting and Delete messages. So in the Delete mode, Fetch and Delete work together in the Keep mode. Only Fetch works not Delete. So the POP3 protocol provides you with a basic facility to access mail from a remote mail server to read mails into a machine so that you can see them. But if you look at the mail clients that are available today it can be Outlook explicit, it can Hotmail, and it can be Rediff Mail. All of them are based on the same technology. You will find that they provide. Many more facilities than just reading the mails one at a time. So there has to be something more supported by the mail access protocol than what POP3 provides and in fact IMAP provides you with the answer.

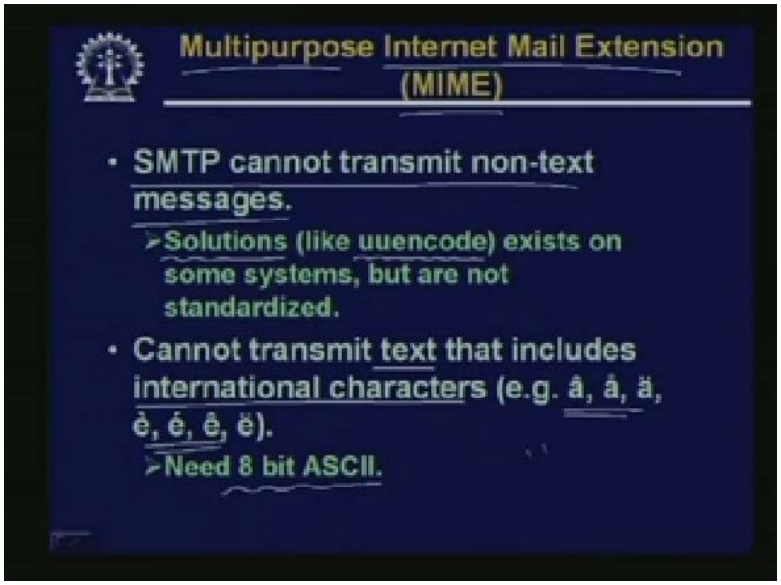
(Refer Slide Time: 40:47)



So in addition to the capabilities provided by POP3, it has the following extra features. First an user can simply check the email headers. See in POP3, you did not have an option, you have fetch the entire mail to your machine and then view them. But using IMAP you can simply fetch the headers of the emails. You just view the headers which one you like you just click on the header. Now the mail will get transferred and you can view it. This is one option; the second is that the user can also search the contents of the email for a specific string. Suppose I want to look only at those mails which have a specific string as part of its body. So I can even do that in IMAP and you can manage mailboxes on the mail server which was not possible in POP3.

So a user can create multiple mail boxes delete a mailbox rename a mailbox or also can create a hierarchy of mailboxes. Like here what I mean is that if you just recall how the typical mail you can say mail utility systems work it can be Outlook, it can be Yahoo, Rediff. See each of these mail servers or let us call it mail systems. They have a facility for the user to specify folders and subfolders. You can manage your mails in terms of multiple folders you can move a mail into a folder; you can delete mails from a folder and so on. So these kind of folder management is directly supported by IMAP because IMAP supports it. Modern day email systems also have a facility to support in the indeed support this kind of you can say management of folders through the IMAP capability.

(Refer Slide Time: 43:05)



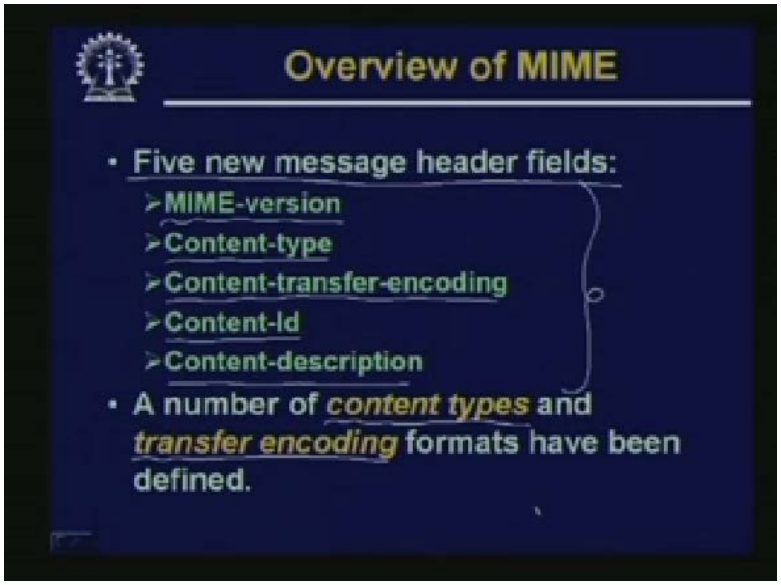
Now we talk about Multipurpose Internet Mail Extension or MIME. Let us try to understand why we have to go for another protocol? Because using SMTP you can send mails using POP3 or IMAP you can read mails on to a machine. The main problem we have till now is that the SMTP protocol has a facility of transmitting only pure text messages, simple text messages which are encoded as 7-bit ASCII. But anything beyond that SMTP is incapable of handling. So you must have a mechanism through which these kinds of non-text documents can be handled. So the first thing is that this SMTP cannot transmit non-text messages. There are some solutions which were developed earlier as adhoc solutions. For example in some UNIX system there was a command called uuencode. This converted a non-text messages into a text equivalent. But it was not accepted as standard. This existed only between UNIX systems. When you want to transmit something between UNIX systems you could do this moreover if you have text that includes international characters like this special symbols which are typically encoded in 8-bit ASCII. Here also you are your simple SMTP will not be allowed; will not be able to handle this kind of text. So for these kind of problems, the protocol MIME was developed in fact MIME can handle something more like.

(Refer Slide Time: 45:07)

Many servers may reject mail over certain size. So the MIME protocol also has a facility of breaking mails into smaller pieces. Moreover some SMTP implementation do not adhere to standards like, even if it is a text, some SMTP implementations will remove actual word spaces like tabs and other things, replace them with single space. So your mail formatting will get lost when it reaches its destination. So MIME tries to keep this mail formatting intact in addition. Now let us try to understand what is MIME is it a new protocol in the same sense that SMTP is? See SMTP is a mail transfer protocol it is a program which is running as a server the client can contact the server and can start a message exchange date.

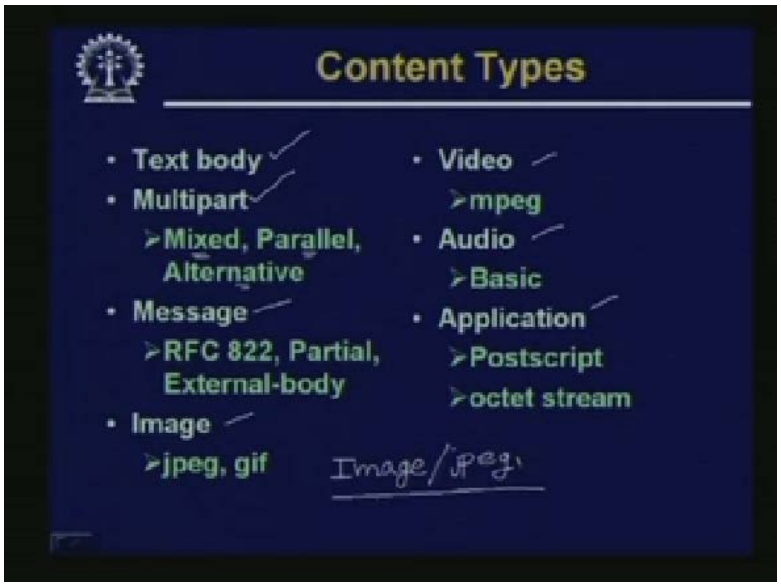
But you remember MIME is a little different from SMTP in this regard. MIME is not a program which is running as a server and someone can contact with MIME and can send and receive something. MIME is not running as a server as a Daemon, rather MIME acts as translator. MIME sits on top of SMTP, if it is non-text mail MIME will try to translate it in some way into an equivalent text version along with relevant information. So that the final receiver can decode it back and there it can forward the message, the translated message to SMTP which can transfer it. This is the basic idea behind MIME. Now in order to have we must have some additional information.

(Refer Slide Time: 46:55)



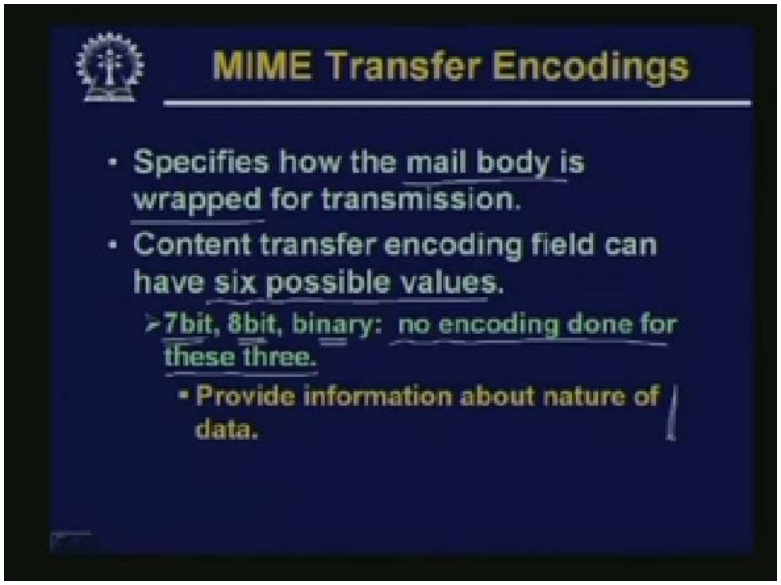
In the translated version I told you, so these additional information are present in the form of 5 new message header fields in the mail messages. There is something called MIME-version, Content-type, Content-transfer-encoding, Content-id, and Content-description. They have their own medium; without going into details of these I shall some examples little later. See these content type actually means what type of content it is? Whatever is following? Whatever you are transmitting is it an image, is it a word document, it is a PDF file? What it is so it specifies the type of the content so you can have types answer subtypes? For example it is an image subtype is it a jpeg image. Sib type can also be a gif image so on. This transfer encoding tells you that what kind of translation you have done. There can be more than one choice available. So the receiver must know about the translation you have done. So that it can be basically translate it back to the original form.

(Refer Slide Time: 48:14)



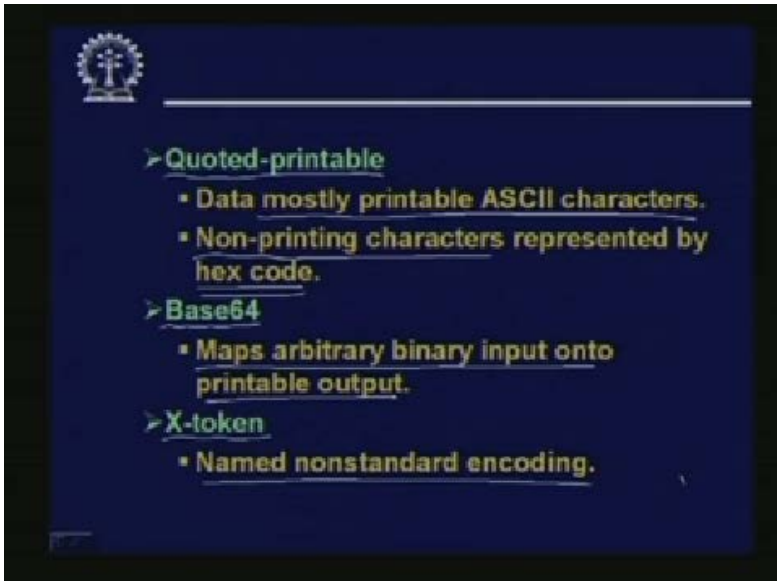
So some of the context types some of content type examples are given here. It can be a Text body; it can be a Multipart in indicating that the mail has multiple parts with the subtypes being Mixed Parallel Alternative Message. Several subtypes Image, Video, Audio, Application and there are several others this is an example the way it is written. For example Image, you give this the type a slash followed by the subtype. This is how you specify the type and subtype.

(Refer Slide Time: 48:50)



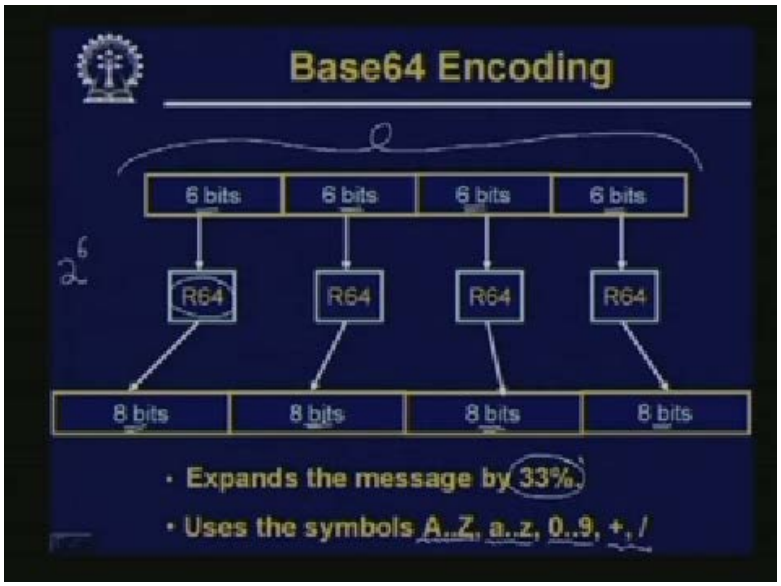
MIME Transfer Encoding specifies how the translation has taken place which in technical term is called how the mail body is wrapped for transmission. Because mail body has to be translated or transformed somehow into a text form. So that SMTP can transmit it this is mandatory. Content transfer encoding can have 6 possible values. Out of them 7 bit, 8 and binary these are special cases. There are no separate encoding required for these because these header, these content transfer encoding type directly tells you what kind of information it is. If you want to send them in plain binary you simply send them 7 bit, 8 bit similar.

(Refer Slide Time: 49:45)



There is one method called Quoted-printable or quote-printable. Here you use this for documents which contain mostly simple ASCII characters with a few non-printable characters. These non-printable characters can be represented by a quote followed by their hexadecimal equivalent. That is why these are called quote-printable if you have a special character give a quote followed by 2 digit hexadecimal equivalent. Base 64 is perhaps the most widely used technique for transforming arbitrary non-text information into a form which can be handled by SMTP. So it maps arbitrary binary input into printable output we shall see how this works and X-token is a method where the user can specify its own nonstandard encoding method.

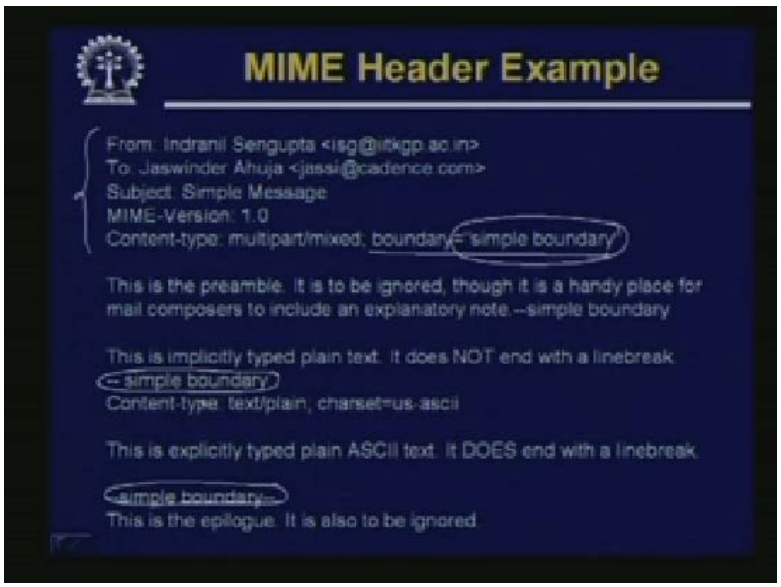
(Refer Slide Time: 50:47)



Now to talk about the Base 64 encoding method, it works like this. Base 64 encoding method can transform any non-text mes method. Message into its equivalent text message. Now the

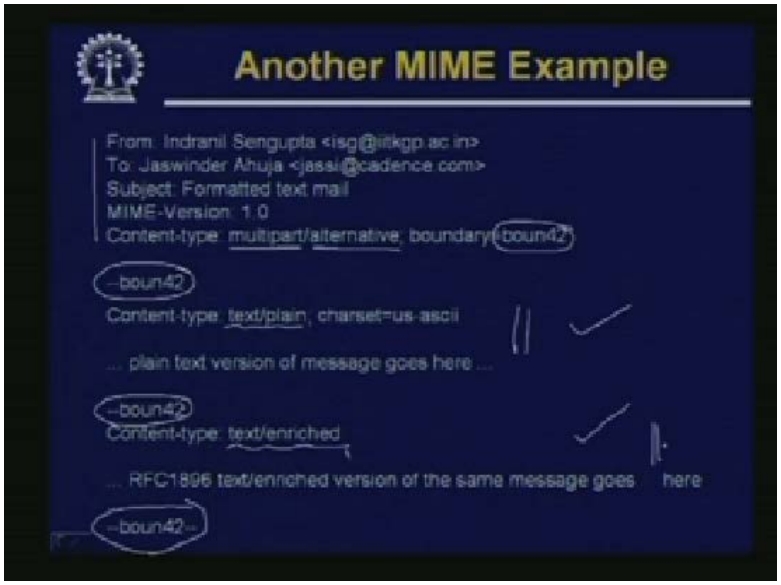
way it works is like this, it takes 3 bytes of the input message at a time. It breaks up into 4 chucks of 6 bits. So the 3 bytes of message is broken up into 6 bits each. Each of these 6 bits are expanded into 8 bits through a simple table look up process. The table look up process is called Are 64 it works like this. See in 6 bits you can have 2 to the power 6 or 64 combinations. So from 0 0 0 0 0 0 to 1 1 1 1 1 1, there are 64 combinations. You have a table which says that for the first 26 of these you replace them by the by the ASCII code of the letters capital A to capital Z. For the next 26 we replace them by their code of small a to small z for the next 10. The digits 0 to 9 and for the last 2, plus and slash, these are the 64 characters. So each of these 64 combinations you encode them as one of 64 ASCII characters. This is how you encode it and replace each ASCII character by its 8 bit ASCII equivalent. So 24 bit gets translated into 32 bits. So this means an expansion in message size by 33 percent.

(Refer Slide Time: 52:36)



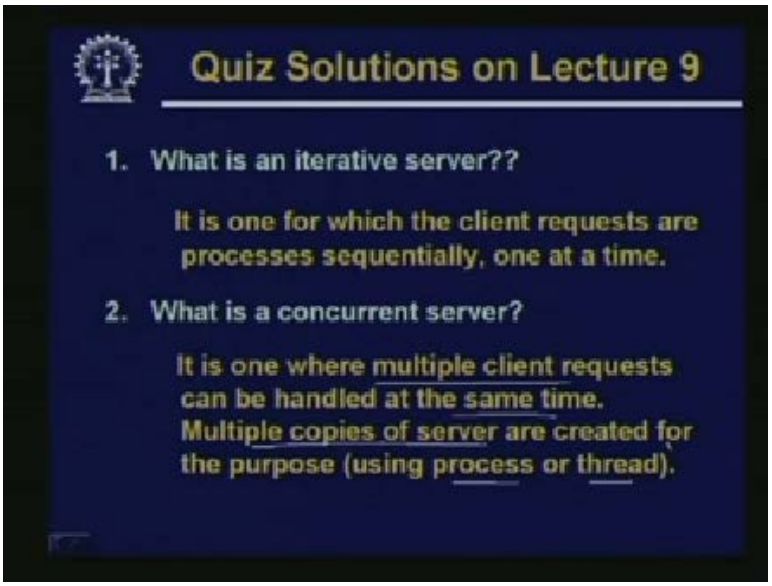
So some examples of MIME header. So just you can look there are some header information. There is a boundary which contains a string. The string wherever it occurs it indicates some boundary in the mail message. There can several such parts in the MIME header.

(Refer Slide Time: 53:02)



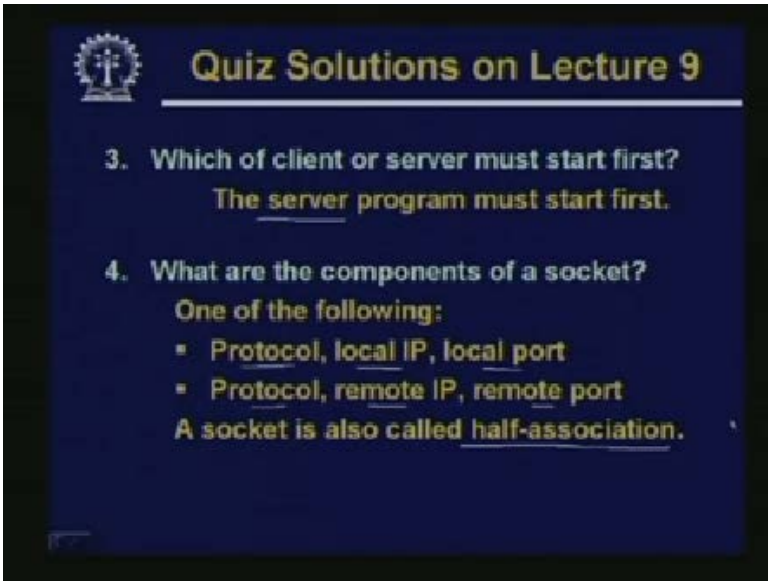
This is just another example. Here are also there are some header, here the boundary name is boun 42. Here there is a boundary, here there is boundary, here there is boundary. So the Content-type for example, multipart alternative means there are multiple parts in the content and alternative means the receiver can choose any one of these two alternatives. Depending on the capability the first one says Content-type text plain. The second one says text enriched so if your receiver mail client can display rich text there is a second alternative can be chosen. Otherwise the first alternative now I suggest you can look at the headers of some of the MIME encoded messages to understand more about how the encoding and the attachments works? The attachments when you send with a mail they get encoded in the same way. Now with this we come to the end of today's lecture. Let us now quickly look at the answers to the quiz questions which were posed as part of the last lecture. Solutions to quiz questions on lecture 9. The questions were:

(Refer Slide Time: 54:18)



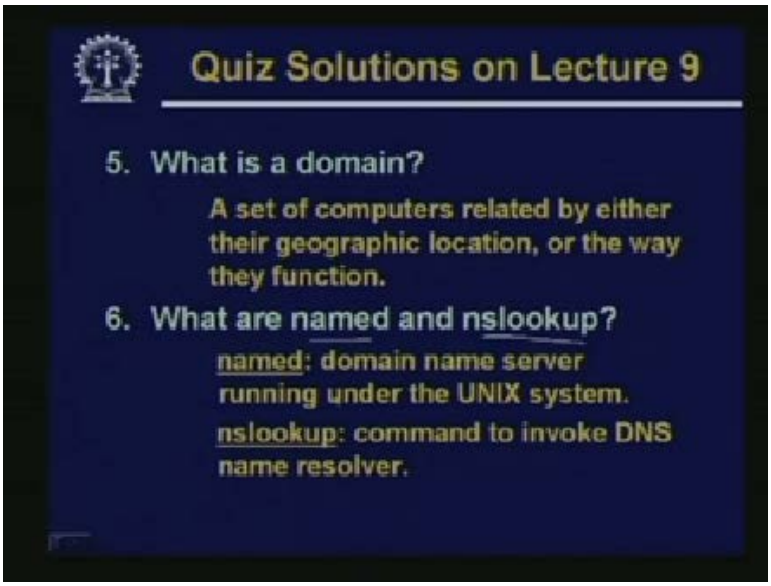
What is an iterative server?
It is one in which the client requests are processed one by one sequentially.
What is a concurrent server?
Here multiple client requests can be handled at the same time. Multiple copies of the server are created using process system call like (54:39 word not clear) or you can also use a thread for this purpose.

(Refer Slide Time: 54:43)



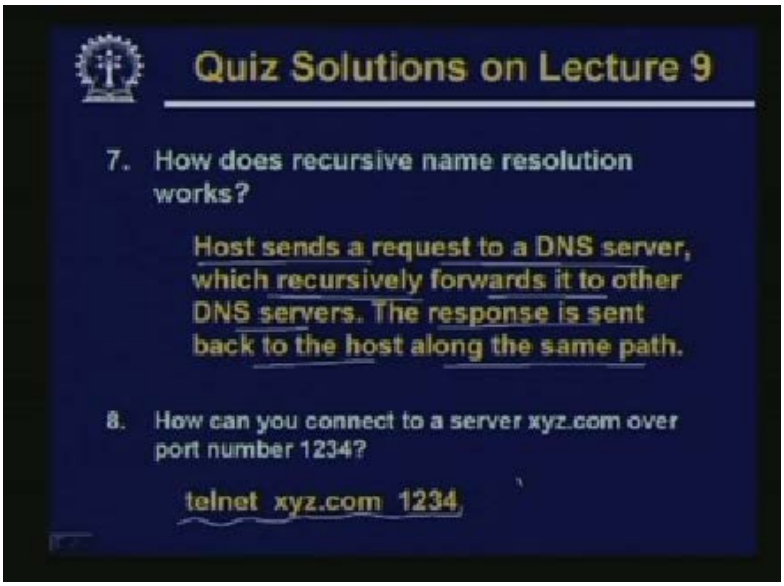
Which of client or server must start first?
Obviously the server.
What are the components of a socket?
It will be Protocol local IP local port or Protocol remote IP remote port it is also called half-association.

(Refer Slide Time: 54:58)



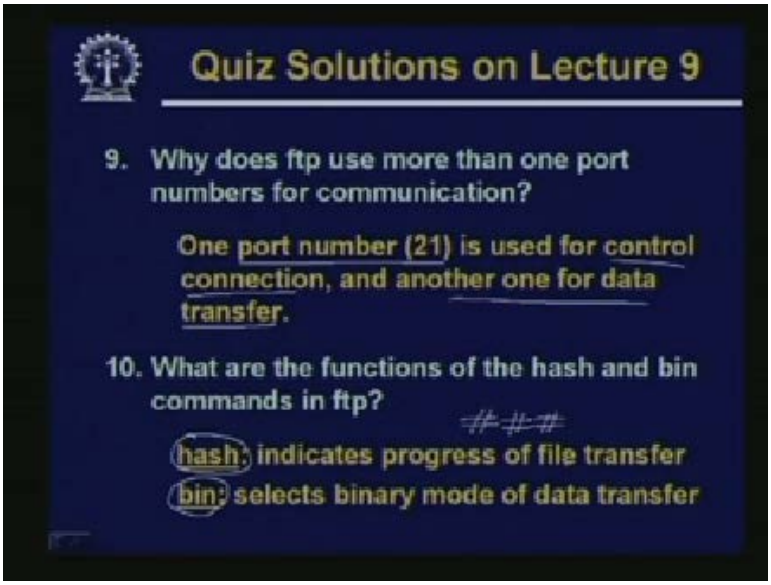
What is a domain?
It is a set of computers which are related either by their geographic location or by the way they function.
What are named and nslookup?
named refers to the domain name server which runs on the UNIX system and nslookup is a command which invoke the DNS name resolver through nslookup you can actually translate a name into an address.

(Refer Slide Time: 55:23)



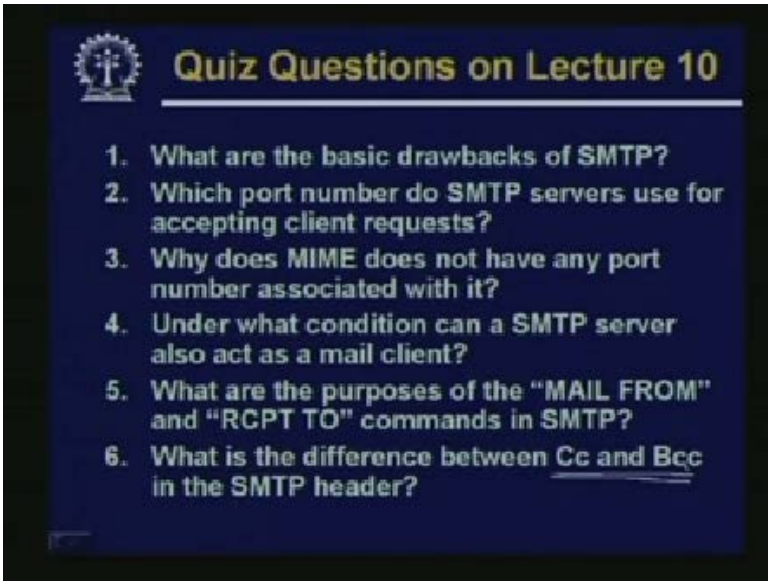
How does recursive name resolution works?
Host sends a request to a DNS server. This DNS server recursively forwards the request to other DNS server the responses are sent back to the initial host along the same path. (55:42 word not clear)
How can you connect to a server xyz.com over port number one twenty 1234?
By a command like this. Telnet server name followed by port number.

(Refer Slide Time: 55:54)



Why does ftp use more than one port numbers?
The first port number 21 is used for control connection and the other one is used for actual data transfer.
What are the functions of the hash and bin commands in ftp?
This hash indicates the status of the file transfer. It displays some hash symbols. Continuously each hash indicates a chunk of data transfer either 2 kilobyte or 4 kilobyte or 8 kilobyte depending on the system and bin selects that you are trying to transmit a non-text file a binary file. Now some questions on today's lecture.

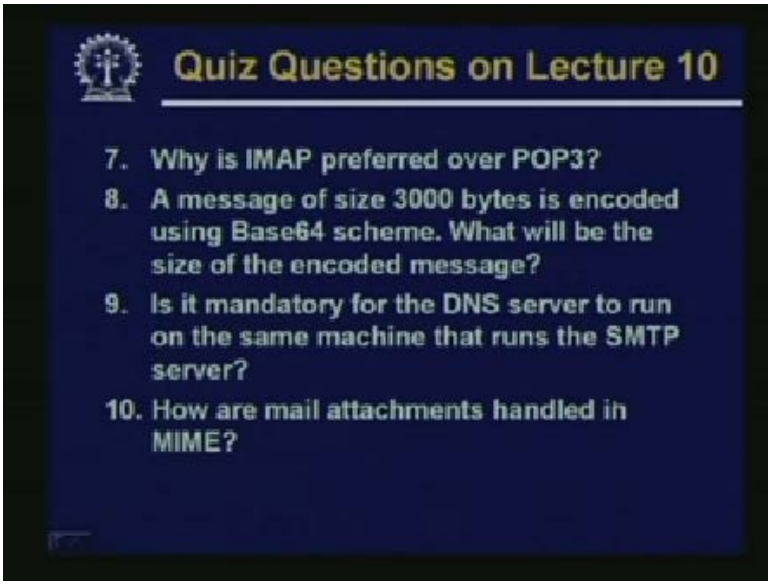
(Refer Slide Time: 56:35)



What are the basic drawbacks of SMTP?
Which port number do SMTP servers use for accepting client requests?
Why does MIME does not have any port number associated with it?

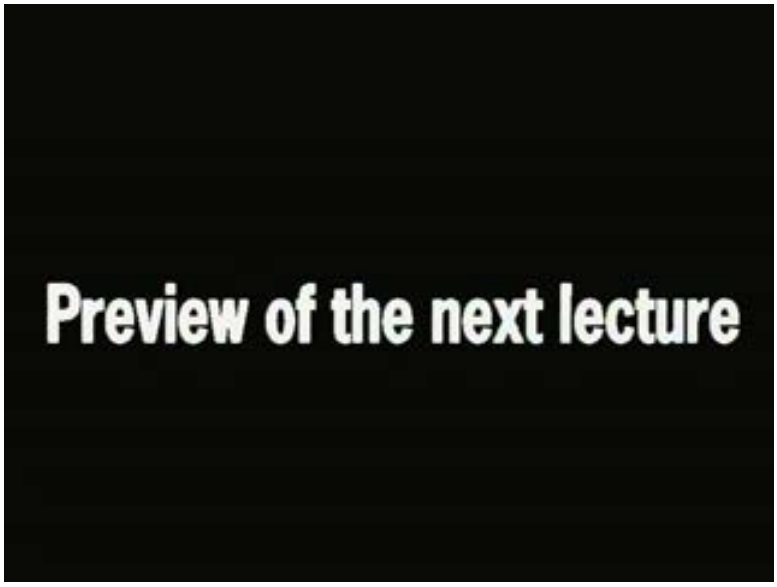
Under what conditions can a SMTP server also act as a mail client?
What are the purposes of the MAIL FROM and receipt TO commands in SMTP?
What is difference between Cc and Bcc in the SMTP header?
This is something I suggest you can look up and find out the solution.

(Refer Slide Time: 57:07)



Why is IMAP preferred over POP3?
A message of size 3000 bytes is encoded using Base64 scheme. What will be the size of the encoded message?
Is it mandatory for the DNS server to run on the same machine that runs the SMTP server?
How are mail attachments handled in MIME?
Now with this come to the end of today's lecture. In our next lecture we shall be starting our discussion on the World Wide Web which is yet another very popular application that we have on the internet. Thank you.

(Refer Slide Time: 57:48)



Preview of next lecture.

(Refer Slide Time: 57:53)

World Wide Web Part- I

In the last class we are talking about electronic mail which is one of the most widely used applications on the internet. Today we will be starting our discussion on another very popular application which i am sure all of you are familiar with. You must have you must have used it in practice and it is the World Wide Web. Now this World Wide Web has become so popular that today World Wide Web and internet, they are used synonymously. Sometimes this World Wide Web or www is meant to be referring to the internet or vice versa so this is an impact of the popularity of this www.